UNLIMITED

RSRE
MEMORANDUM No. 4263

ROYAL SIGNALS & RADAR ESTABLISHMENT

ON THE PROBLEM OF SECURITY IN DATA BASES

Author: Simon Wiseman

PROCUREMENT EXECUTIVE,
MINISTRY OF DEFENCE,
R S R E MALVERN,
WORCS.

RSRE MEMORANDUM No. 4263

UNLIMITED

CONDITIONS OF RELEASE

0038157 BR-109899

.... U

.... Y

Reports quoted are not necessarily available to members of the public or to commercial organisations.

ROYAL SIGNALS AND RADAR ESTABLISHMENT

Memorandum 4263

Title: On The Problem of Security in Data Bases
Author: Simon Wiseman
Date: January 1989

Abstract

This paper addresses the fundamental problems of providing multi-level secure data base management systems. It is shown that the *Inference problem arises because data is used as an* addressing mechanism. Existing solutions are described, but these are shown to have unfortunate consequences for the data base user and so alternatives are suggested. The problems of Aggregation and Deductions from real world knowledge are also considered.

Introduction

A data base management system (DBMS) gives the users a view of their data which is independent of the way in which it is actually stored. This allows the physical structure of the data base to be modified to reflect changes in requirement, without affecting the applications which use it (at least this is the hope, in practice not all products offer this!). A particularly important consequence of this high level view is that the containers of data (eg. records) are addressed by content rather than direct physical address.

A multi-level secure (MLS) system is one which handles information of various classifications and provides some degree of assurance that information retrieved from the system is properly classified. The system's users generally have a variety of clearances which should deny them access to some of the information.

The development of a multi-level secure data base management system faces a particular problem which arises because data is accessed by content, but the complexity of tools provided by data bases is an added complication. This paper exposes this fundamental problem and describes solutions which have been proposed for its solution. These, however, are shown to have unfortunate consequences for the data base user. An alternative solution is proposed which allows for a balanced set of measures to be employed to solve the security problem of a particular secure data base.

Covert Channels

It may seem strange to commence a discussion about the fundamental problems of data base security with a discussion about covert channels, but it transpires that they are the root cause of the most important data base security problem.

Communication can occur whenever a resource, such as memory or CPU time, is shared. A user can send information to others by modulating the state of the shared resource. Other users who can detect changes in the state can receive the information. In a secure system communication is controlled to enforce the system's security policy.

Unfortunately, whenever a protection mechanism is employed to prevent information flowing in undesirable ways, a new communication medium is made available, namely the mechanism itself. Information can be transmitted by modulating the protection mechanism's state and is received simply by distinguishing between different states. Such channels are significant problems in the design of secure systems. They are clearly not intended to be used for communication and so are called covert channels.

Note that it will always be possible to distinguish at least two states in a protection mechanism because, by their very nature, they will either allow or prevent an access. It may be possible to encapsulate the channel inside another protection mechanism, but in effect this is just pushing the problem back a stage. Eventually a protection mechanism will remain in the system which can be used for covert communication.

Ultimately, the only solution is to prevent sensitive information from being encoded in the channel. This may be done by ensuring either that the protection state can never be changed or that sensitive information is not encoded in any changes. The former solution is simple and can be used to good effect, but can be rather limiting on functionality. The latter requires the ability to limit changes to those entities which are trusted to make changes without encoding any sensitive information. One example of this is the Trusted Path to the human user [Wisemanetal88].

The weakest form of protection mechanism available is an addressing mechanism. However, it should be noted that even here covert channels are present, with information being encoded in the existence of the entity being addressed. The next section explains how covert channels through addressing mechanisms are the root cause of the main data base security problem.

Inference

A computer system can be modelled as a number of containers of data [Terry&Wiseman89]. In a conventional system, data is accessed by addressing the container which holds it. For example, a read operation may ask for "the data contained in record number 42".

A data base differs from a conventional system in one important respect. The requests for information do not directly name the containers which are to be accessed. Instead the containers are addressed by content rather than by name. For example, a data base query may ask "give me the contents of all records which contain the word 'fighter'".

Obviously in an MLS system some controls must be applied to accessing data. The usual mechanism is to attach security classifications to all the containers in the system and to ensure containers may only be accessed by those users whose clearance permits this.

Now consider the covert channels which exist in a conventional system where security is enforced by labelling containers. There are in fact two controls being imposed upon accesses. The first control is simple addressing, a very weak form of protection, which ensures data is not accessed unless it is in a container which exists. The second control prevents an access if the user's clearance is insufficient. Since there are two different access controls there are two covert channels. Information may be covertly encoded either in the existence of containers or in their classification. Note that the data itself is also a communication channel, but is not covert because its use is intended and is governed by the labelling mechanism.

The first of these covert channels can be closed by preventing users creating and deleting containers and the second by preventing them from changing the classification of existing containers. If these solutions prevent the system from having essential functionality, which is highly likely, the only alternative is to use the Trusted Path to ensure that no sensitive information is encoded in the changes.

If neither banning all changes nor restricting them to the Trusted Path are acceptable solutions, the only course is to find alternative protection mechanisms. For example, making the classification part of the basic addressing mechanism closes the covert channel through the creation of containers. This, however, may itself have an impact on the functionality of the system.

Now consider the covert channels which arise in a secure data base that uses labelled containers to enforce security by preventing the user from seeing the result of a query if it is classified too high. Here there are two communication channels, one through the data and one through the containers' classifications. Obviously the classifications provide a covert channel and can be dealt with as in conventional systems. Surprisingly however the data is also a covert channel, even though it appears to be closed by the use of labels.

The covert channel through the data arises because labelling does not prevent the data from being addressed, only from being seen directly. That is, the channel arises through the data's role in addressing rather than from it being accessed. Thus the user of such a data base can pose a query which addresses some sensitive data. If the data does not exist an empty set will be returned, whereas if it does exist the user will be denied access to the result. From this, however, the user can readily infer that the facts given by the data are true.

This then is the data base inference problem. It is essentially a covert channel which is driven by users innocently storing their information in the data base. The information can be received by other users addressing the data using the query mechanisms and inferring its existence by being denied access to it.

For example a user with a low clearance may ask "give me the contents of all containers which contain the fact that flight 127 is carrying bombs to The Front". If the data base contains such a fact, and it is highly classified, the user will be denied access to it. If the fact is not in the data base the result would be an empty set. Unfortunately the user can now infer whether flight 127 is indeed carrying bombs to The Front, even though the data base didn't actually reveal this directly.

It may be practical in many systems to stop the covert channel through the labels by insisting that it be driven only from the Trusted Path. However, it is obviously impractical to stop the covert channel through addressing by preventing it being driven, because this would mean no information could be stored in the data base. The only alternatives are to find a way of preventing the information from being received or to use a different addressing mechanism.

A Solution - Polyinstantiation

One technique which has been proposed for providing a MLS DBMS is called Polyinstantiation [Denningetal87]. This avoids the covert signalling channels, and hence solves the inference problem.

The idea behind the technique is that users are prevented from being influenced by any information which they are not cleared to see. In particular, queries do not 'see' any information classified higher than the user's clearance, while updates, insertions and deletions cannot be 'seen' by users with lower clearances.

This is fine except that updates may cause the data base to enter an inconsistent state. Different and contradictory facts about something have to bc recorded, to ensure that the state observed by lowly cleared users is not affected by the actions of highly cleared users.

The effect this approach has on a Relational Data Base is to require that more than one record in a relation be associated with a particular key. To ensure that the data base is still normalised the relation is treated as if its key field were extended to include the classification of the record. Thus records with the same key, but different classifications, may appear in a relation, hence the name 'Polyinstantiation'.

Whenever a new record is inserted into a relation, its classification is appended to its key field. Whenever a query is made, only those records whose classification is dominated by the clearance of the user are inspected. Thus the effects of entering highly classified information are hidden from lowly cleared users.

The goal of Polyinstantiation is that the state of the data base as seen by lowly cleared users cannot be altered by highly cleared users. Therefore lowly cleared users may query the data base and receive out of date information rather than be told they are not cleared to know the answer. In effect, Polyinstantiation gives a secure data base, but at the expense of providing a data base which lies.

The reason Polyinstantiation achieves security is because it combines the classification checks with the addressing mechanism. In effect the user cannot address sensitive data whether it exists or not. If a query asks for sensitive data the reply is in effect "here are the facts you requested and are cleared to see, though they may have since been augmented, rescinded or altered by users cleared higher than you". Therefore the users cannot deduce the truth of a sensitive fact since they will never be told their clearance is insufficient.

An Example of Polyinstantiation

Take as an example a Polyinstantiating data base which records cargo and destination for aircraft. Suppose an officer (and a gentleman) wishes to send bombs to the forces at The Front. He queries the data base and discovers that aircraft F127 is suitable and available. He then 'books' that aircraft by recording in the data base that F127 is carrying bombs to The Front. He decides that, for strategic reasons, this fact is Secret and informs the DBMS of this when the new information is entered.

Now suppose the system is also used by the Army Catering Corps to arrange delivery of rations to the troops. This activity is less sensitive than supplying armaments, so the officer in charge is only cleared to Confidential. Wishing to restock forces at headquarters with Champagne, the catering officer queries the data base and finds that aircraft F127 is suitable and available. He is not told that it is already booked because he is only cleared to Confidential and hence, because of Polyinstantiation, his query does not see the secret fact. Therefore the officer goes ahead and arranges for F127 to carry Champagne to HQ.

The data base now contains two facts. One, which is secret, says that F127 is carrying bombs to The Front and the other, which is confidential, says that F127 is carrying Champagne to HQ. The data base is therefore inconsistent. This has arisen because Polyinstantiation caused the data base to lie to the catering officer.

If the data base is not to lie, the catering officer should have been told that he is not cleared to know which aircraft are available. This is because F127 is booked for a secret mission and this fact is used to calculate which aircraft are available.

Continuing the above example, consider what happens if the requests to use F127 were entered in the opposite order. First, the catering officer would arrange for F127 to carry Champagne to HQ, a fact which he says is confidential. Now suppose the officer in charge of armaments decides it is vital that bombs be delivered to The Front, yet he finds that no aircraft are available. He then elects to use his authority to rescind the order for F127 to carry Champagne and instead orders it to deliver the bombs.

In these circumstances the Polyinstantiating data base prevents the catering officer from seeing the effects of the new orders. He still sees the confidential fact that F127 is carrying Champagne. The data base actually holds the additional secret fact that F127 is carrying bombs and both of these are seen by the armaments officer.

If the data base were not to lie, the catering officer would now see that he is no longer cleared to know about F127's movements. This of course gives the officer knowledge about the secret activities of the armaments officer, but use of the Trusted Path would prevent this from being exploited as a covert channel.

Now suppose that other officers use the data base to receive their orders. The flight crew of F127 are cleared to confidential, because they do not need to know about the cargo they are transporting. Therefore the data base tells them they are going to HQ, because the fact that they should be going to The Front is secret. Note that the data base has lied and the crew are about to make a big mistake. A more desirable course would be for the data base to tell them they are not cleared to know the destination, which while inconvenient is at least safe.

Now suppose the ground crew also use the system to know what cargo to load. They are cleared to secret, because they actually have to handle classified goods. On consulting the data base about F127 they can see it is carrying bombs (assume that time stamps are used to determine that the Champagne order has been superseded).

Thus the ground crew load the correct cargo, but the flight crew deliver it to the wrong place. This arises because Polyinstantiation prevents all information flow from highly cleared to lowly cleared users, not just the undesirable flows. The results are unacceptable for an operational system and is a fine example of security controls preventing a job from being done properly.

Classification by Content

Another technique which has been proposed for achieving security in data bases, is essentially classifying containers by their content rather than by explicit security labels [Wilson88]. To do this, every fact which could conceivably be stored in the data base must be assigned a classification. This would be done by specifying a set of classification rules.

When users pose queries, the system checks the classification rules to decide what classification of data is liable to be accessed while the result is calculated. If the user is not cleared to access the data, whether it exists or not, the query is rejected. Thus, by combining the classification check with the addressing mechanism, the covert channel which allows inferencing does not arise.

This solution is secure and has the advantage that it does not lie. Unfortunately there are a number of drawbacks, in particular a lowly cleared user may find it difficult to get any answer from the data base. This is because the possibility of there being highly classified information in the data base prevents lowly classified information being divulged to lowly cleared users.

To make such a system usable, all data base accesses can be made through views, which are a mechanism for subsetting the facts in a data base. For example, one view of a data base may be those records which are classified lower than a particular classification.

Using views, the data base users can constrain the system to consult just a subset of the data base when computing the result of a query. Thus a query on a suitable view will only look for data that the user is cleared to see and so there is no possibility of rejection and the associated inference.

An Honest Secure Data Base

Both Polyinstantiation and Classification by Content suffer from much the same problem. Users can never make a query to which they receive the answer "you are not cleared to know this". This is because Polyinstantiation lies and returns false information and Classification by Content refuses to answer queries which could conceivably give such an answer.

These solutions close the signalling channel that arises because data is addressed by content and so prevent any inferences being made. The drawback is that users are denied access to certain lowly classified information. That is they are not allowed to know that the data base contains facts that they are not cleared to see, which is a fact that is usually unclassified.

In some applications it may be acceptable for users to be in a position to infer small amounts of information which strictly they are not cleared to know. They may be trusted not to misuse it, even though strictly they should not possess the information. The system's owners may consider such an approach to be acceptable so long as the event is properly recorded, allowing for later audit.

Another possibility is that most users have clearances which permit them to view all the information in the data base. In such a system the problem is not that the users may make inferences, but that Trojan Horses in the software they run may use inferencing to downgrade information. This would result in information being inappropriately cared for. Therefore, in a system which uses high water marks [Cummingsetal87] to constrain the activities of Trojan Horses, rather than having users work at fixed session levels, the mechanism must account for information gained by inference.

Thus the choice of a suitable access control mechanism is highly application dependent. It is even likely that different mechanisms will be required for different parts of an application's data base. Therefore, a general purpose multi-level secure data base must offer a range of mechanisms and a way of showing that, in a particular application, they work together to provide security.

There are several mechanisms we have available to choose from. First we can control access to the collections of records as a whole, for example a relation in a relational data base. Note these are not addressed by content. it is the records inside them that are, so there is no inference problem.

Second, we may restrict the ability to use all fields of a record in the content based addressing. This would prevent users from being able to infer information about some of the fields, while allowing them to attempt to access the data. Note though that this option is really a generalisation of the first, since a user who is unable to use any fields for addressing cannot access any of the records.

Third, when untrusted software makes a query which allows it to infer some highly classified information, it can be eliminated. Effectively this amounts to shooting Trojan Horses. This does not however prevent the human user from making these inferences, and, since it is presumably impractical to shoot the users, it will be necessary to audit the event and trust the user to handle the information correctly.

The fourth option is to make the data base lie, by using Polyinstantiation, or refuse to answer, by using Classification by Content. If these mechanisms are applied selectively they may be entirely satisfactory.

Deductions using Real World Knowledge

The users of a data base can deduce additional information by employing knowledge which is not obtained from the data base. They achieve this by making logical deductions from facts obtained from the data base using information available in the real world. This is often called 'inference' but, to avoid confusion with inferences made using the covert channel, the term 'deduction' will be used. Note that the extra information may be stored in the data base, but the user has either elected not to obtain it or is not able to retrieve it because of the system's security controls.

For example, suppose a data base recording flight information reveals the unclassified fact that a flight is carrying an item of weight 100Kg. The nature of the item is highly classified and is not revealed. Suppose the only kind of item likely to be carried which weighs 100Kg is a bomb, then the lowly cleared uer can put two and two together and deduce the highly classified fact that the flight is carrying a bomb.

Obviously the computer system cannot prevent the use of real world knowledge which is outside its influence. However the system design process must ensure that the system's owner is aware of the attacks which can be made in this way.

It should be noted that using real world knowledge to deduce highly classified information from lowly classified facts is not a problem peculiar to data bases. These deductions can only be avoided by considering the system as a whole and ensuring that the computer system is appropriately structured. However, inferences which are made via the covert channel through content based addressing is a data base problem, and the following sections discuss various mechanisms which have been proposed to avoid it.

Aggregation

It is possible that an application will wish to store information which is composed of smaller pieces of information. The small pieces on their own may be relatively lowly classified, however when taken together the aggregation of them may be highly classified. The problem a secure system must solve is to allow lowly cleared users to observe some of the small pieces, but not so many that the full highly classified aggregate is made available to them.

For example, it may be Unclassified that F127 is going somewhere, Unclassified that some flight is carrying bombs and unclassified that some flight is going to The Front. However, it may be Confidential to know either that F127 is carrying bombs or that some flight is carrying bombs to the Front or that F127 is going to The Front. Knowing that F127 is carrying bombs to The Front may then be Secret.

In this example, a user cleared only to see Unclassified information could ask the data base what flights are busy and be told that F127 is going somewhere. However if the same user then asked what kinds of cargo are being moved they must be denied access to the answer. This is because they would otherwise possess the aggregate fact that F127 is carrying bombs, which is Confidential.

The difficulty is that information may be retrieved gradually over a period of time. The check that clearance is being exceeded obviously cannot be made at the start of this process, because at this stage the highly classified aggregate has not been revealed. The check must occur later when an attempt to obtain the last piece of a highly classified aggregate is made. This implies that aggregation can only be controlled by a protection mechanism based on a history of accesses.

The aggregation problem is compounded by the ability of users to gain information by making deductions using knowledge obtained from the real world. This will inevitably bypass any aggregation control mechanism.

Clearly it is impractical to keep a complete history of all information which the data base has divulged, both directly and through inference, to its users and to consult this every time queries are made. However, the size of the history could be reduced by removing entries which are made redundant through updates and deletions and by reducing the precision of the mechanism.

Even so, a successful solution to the aggregation problem looks unlikely for performance reasons. However, instead of despairing we must closely question the need for a protection mechanism like aggregation. In particular we must look closely at the requirements of real systems in terms of the threats against them.

The aggregation problem is not solely the prerogative of data base systems. It can occur in paper world where a file containing many reports may be secret, even though each report is only confidential. The individual reports may still be seen by people who are only cleared to confidential, but the collection must be locked away in a safe that is strong enough for secrets. This is because loss of the entire collection would be extremely damaging, but the loss of just one would be less so. A data base encourages the user to give structure to the information and thus to its classification. Therefore the problem arises more often in data base systems than in conventional systems.

Thus in practice aggregation is used to increase the protection given to information, that is to give higher assurance that it will not be lost, rather than to restrict its dissemination. This suggests that aggregation is used to address issues of assurance in paper world confidentiality controls and other solutions may be appropriate in a computer system. While saying this does not offer a solution to the aggregation problem, it does show that the requirement for solving it must receive closer inspection.

Conclusions

The particular problem with secure data bases is that, unlike conventional systems, the containers which hold the classified data are accessed by content rather than by name. The use of the data by the addressing mechanism is not controlled by classification labels. Thus a covert channel exists and it is possible to infer or guess highly classified information by being denied access to it. This is the inference problem.

The solutions to this problem which have been offered up to now are Polyinstantiation and Classification by Content. These techniques have the unfortunate property of making the data base either lie to or withhold information from, its lowly cleared users. We do not believe this is acceptable.

In common with conventional systems, data bases also suffer from covert channels which exploit the ability to encode information in classification labels and the ability to use real world knowledge to deduce highly classified information.

Data bases also encourage the use of structured data, allowing aggregations of lowly classified data to be highly classified. Thus secure data bases suffer from problems where highly classified information can be gradually pieced together from lowly classified data. This aggregation problem, which is not peculiar to data bases, appears to be impossible to solve satisfactorily because it must depend upon maintaining and consulting a history of accesses.

We propose to provide a number of mechanisms which can be applied as appropriate according to the needs of an application. The construction and analysis of a security model ensures that these mechanisms are applied consistently and completely. Thus there is a role for Polyinstantiation and Classification by Content, but not as system wide access control mechanisms and only as a last resort.

References

P T Cummings, D A Fullam, M J Gosselin, J Picciotto,
J P L Woodward & J Wynn
Compartmented Mode Workstation: Results through Prototyping
Procs. IEEE Symposium on Security and Privacy,
Oakland, California, April 1987

D E Denning, T F Lunt, R R Schell, M Heckman & W Shockley
A Multilevel Relational Data Model
Procs. IEEE Symposium on Security and Privacy,
Oakland, California, April 1987

P F Terry & S R Wiseman
A 'New' Security Policy Model
Procs. IEEE Symposium on Security and Privacy,
Oakland, California, May 1989

J Wilson
Views as the Security Objects in a Multilevel Secure Relational
Database Management System
Procs. IEEE Symposium on Security and Privacy,
Oakland, California, April 1988

S Wiseman, P Terry, A Wood & C Harrold
The Trusted Path between SMITE and the User
Procs. IEEE Symposium on Security and Privacy,
Oakland, California, April 1988

DOCUMENT CONTROL SHEET

Overall security classification of sheet ............UNCLASSIFIED...........................

(As far as possible this sheet should contain only unclassified information. If it is necessary to enter classified information, the box concerned must be marked to indicate the classification eg (R) (C) or (S))

1. DRIC Reference (if known)	2. Originator's Reference Memo 4263	3. Agency Reference	4. Report Security Classification U/C

5. Originator's Code (if known) 7784000	6. Originator (Corporate Author) Name and Location ROYAL SIGNALS & RADAR ESTABLISHMENT ST ANDREWS ROAD, GREAT MALVERN, WORCESTERSHIRE WR14 3PS
5a. Sponsoring Agency's Code (if known)	6a. Sponsoring Agency (Contract Authority) Name and Location

7. Title

On the problem of security in data bases.

7a. Title in Foreign Language (in the case of translations)

7b. Presented at (for conference papers) Title, place and date of conference

8. Author 1 Surname, initials Wiseman S	9(a) Author 2	9(b) Authors 3,4...	10. Date 1989.01	pp. ref. 10

11. Contract Number	12. Period	13. Project	14. Other Reference

15. Distribution statement

Unlimited

Descriptors (or keywords)

continue on separate piece of paper

Abstract

This paper addresses the fundamental problems of providing multi-level secure data base management systems. It is shown that the Inference problem arises because data is used as an addressing mechanism. Existing solutions are described, but these are shown to have unfortunate consequences for the data base user and so alternatives are suggested. The problems of Aggregation and Deductions from real world knowledge are also considered.